# A Survey of TCP over Mobile Ad Hoc Networks

Ahmad Al Hanbali, Eitan Altman, Philippe Nain

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# A Survey of TCP over Mobile Ad Hoc Networks

Ahmad Al Hanbali, Eitan Altman, Philippe Nain

N° 5182

May 2004

Thème COM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--5182--FR+ENG

# A Survey of TCP over Mobile Ad Hoc Networks

Ahmad Al Hanbali, Eitan Altman, Philippe Nain

Thème COM —
Projets Maestro

Rapport de recherche n° 5182 — May 2004 — 28 pages

**Abstract:** The Transmission Control Protocol (TCP) was designed to provide reliable end-to-end delivery of data over unreliable networks. In practice, most TCP deployments have been carefully optimized in the context of wired networks. Ignoring the properties of wireless Mobile Ad Hoc Networks (MANETs) can lead to TCP implementations with poor performance. In order to adapt TCP to MANET environment, improvements have been proposed in the literature to help TCP to differentiate between the different types of losses. Indeed, in MANETs losses are not always due to network congestion, as it is mostly the case in wired networks. In this report, we present an overview of this issue and a detailed analysis of the major factors involved. In particular, we show how TCP can be affected by mobility and lower layers protocols. In addition, we review the main proposals which aim at adapting TCP to MANET environment.

**Key-words:** Mobile Ad Hoc Networks, TCP, routing protocol, packet loss, power constraint, cross layer.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

# Etat-de-l'Art de TCP dans les Réseaux Mobiles Ad Hoc

**Résumé :** TCP (*Transport Control Protocol*) a été conçu pour fournir un service de transport de bout en bout fiable. Au cours des ans, des versions successives de TCP ont vu le jour, dans le but d'optimiser les performances du protocole. Toutefois, ces améliorations ont été réalisées uniquement dans le contexte des réseaux filaires, comme l'Internet. L'avènement des réseaux sans fil, en particulier les réseaux ad hoc mobiles, a mis à jour des comportements indésirables de TCP, qui dégradent fortement les performances (temps de réponse, délai, etc.). Plusieurs contributions visant à adapter TCP aux réseaux sans fil ont récemment été proposées. L'objectif de cet article est d'identifier les problèmes spécifiques liés à l'utilisation de TCP dans les réseaux sans fil et à faire le point sur les dernières solutions proposées pour y remédier.

**Mots-clés :** Réseaux mobile sans fil, TCP, protocole de routage, perte de paquet, contrainte de puissance.

# 1 Introduction

Mobile Ad Hoc Networks (MANETs) are complex distributed systems that consist of wireless mobile nodes that can freely and dynamically self-organize. In this way they form arbitrary, and temporary "ad hoc" networks topologies, allowing devices to seamlessly interconnect in areas with no pre-existing infrastructure. Recently, the introduction of new protocols such as Bluetooth [1], IEEE 802.11 [2] and Hyperlan [3] are making posssible the deployment of MANETs for commercial purposes. As a result, considerable research efforts have been put on this new challenging wireless environment.

TCP (Transmission Control Protocol) was designed to provide reliable end-to-end delivery of data over unreliable networks. In theory, TCP should be independent of the technology of the underlying infrastructure. In particular, TCP should not care whether the Internet Protocol (IP) is running over wired or wireless connections. In practice, it does matter because most TCP deployments have been carefully optimized based on assumptions that are specific to wired networks. Ignoring the properties of wireless transmission can lead to TCP implementations with poor performance.

In MANETs, the main problem of TCP lies in performing congestion control in case of losses that are not induced by network congestion. Since bit error rates are very low in wired networks, nearly all TCP versions nowadays assume that packets losses are due to congestion. Consequently, when a packet is detected to be lost, either by timeout or by triple duplicated ACKs, TCP slows down the sending rate by adjusting its congestion window). In the case of timeout, TCP enters in the slow start phase by setting its current coggestion window to 1. In the case of triple duplicated ACKs, TCP enters the congestion avoidance phase by halving its current congestion window. Unfortunately, wireless networks suffer from several types of losses that are not related to congestion, making TCP not adapted to this environment.

Numerous enhancements and optimizations have been proposed over the last few years to improve TCP performance over one-hop wireless (not necessary ad hoc) networks. These improvements include infrastructure based WLANs [4, 5, 6, 7], mobile cellular networking environments [8, 9], and satellite networks [10, 11]. MANETs inherit several features of these networks, in particular high bit error rates and path asymmetry, and add new problems that come from mobility and multi-hop communications, such as network partitions, route failures, and hidden (or exposed) terminals.

We classify the main approaches that aim to improve TCP performance in MANETs in two categories: cross layer proposals and layered proposals. In cross layer proposals, TCP and its underlying protocols work jointly. For example, considerable improvements are possible when TCP can differentiate between packet losses due to congestion that should activate the congestion control algorithm, and losses due to the specific features of MANETs. In order to do that, some proposals suggest that when the routing layer detects a route failure, it notifies the TCP sender about a routing failure [12, 13, 14, 15, 16]. Upon receiving this notification, TCP sender enters a *freezing* state. In this state, TCP stops sending data packets, and it freezes all its variables to their current value, such as the congestion window and the retransmission timer. After route re-establishment, TCP sender goes back to the normal state. In layered proposals, the problems of TCP is attacked at one of the OSI layers. For example, in [17] the authors use adaptive TCP delayed ACK to reduce the contention on wireless channel. In [18], the authors propose two link layer techniques, called Link RED and adaptive pacing, to improve TCP performance.

The rest of the report is organized as follows: In Section 2 we discuss TCP's challenges in MANETs environment. In Section 3 we survey papers on TCP performance over mobile wireless multi-hop ad hoc networks. In Section 4 we review the main proposals for improving TCP performance. Section 5 concludes the paper and gives research directions.

# 2 TCP's challenges in MANETs environment

The performance of TCP degrades in MANETs environment, This happens because TCP has been optimized for running over wired networks. In MANETs, TCP has to face new challenges arising from the specificity of these networks: high bit error rates, path asymmetry, network partitions, route failures and route re-establishments, power constraints.

## 2.1 Lossy channels

The main causes of errors in wireless channel are the following:

- Signal attenuation: This is due to a decrease in the intensity of the electromagnetic energy at the receiver (e.g. due to long distance), which leads to low signal-to-noise ratio (SNR).
- Doppler shift: This is due to the relative velocities of the transmitter and the receiver. Doppler shift causes frequency shifts in the arriving signal, thereby complicating the successful reception of the signal.
- Multipath fading: Electromagnetic waves reflecting off objects or diffracting around objects can result in the signal traveling over multiple paths from the transmitter to the receiver. Multipath propagation can lead to fluctuations in the amplitude, phase, and geographical angle of the signal received at a receiver.

In order to increase the successful transmission ratio link layer protocols implement the following techniques: Automatic Repeat reQuest (ARQ)[1], or Forward Error Correction (FEC), or both of them[2]. Further in MANETs, stations may rely on physical carrier-sensing

[1] In 802.11, when a transmitter detects an error, it will retransmit the frame, error detection is timer based

[2] Bluetooth implements both ARQ and FEC on some synchronous and Asynchronous connections

mechanism to determine idle channel, such as in the IEEE 802.11 DCF function. This sensing mechanism generates complex phenomena, such as the *hidden station* and the *exposed station* problems [19]. Before explaining these problems, we need to clarify the "transmission range" term. The transmission range is the range, with respect to the transmitting station, within which a transmitted packet can be successfully received.

A typical hidden terminal situation is depicted in Figure 1. Stations A and C have a frame to transmit to station B. Station A cannot detect C's transmission because it is outside the transmission range of C. Station C (resp. A) is therefore "hidden" to station A (resp. C). Since A and C transmission areas are not disjoint, there will be packet collisions at B. These collisions make the transmission toward B problematic. To alleviate the hidden station problem, virtual carrier sensing has been introduced [2, 20]. It is based on a two-way handshaking that precedes data transmission. Specifically, the source station transmits a short control frame, called Request-To-Send (RTS), to the destination station. Upon receiving the RTS frame, the destination station replies by a Clear-To-Send (CTS) frame, indicating that it is ready to receive the data frame. Both RTS and CTS frames contain the total duration of the data transmission. All stations receiving either RTS or CTS will keep silent during the data transmission period (e.g. station C in Figure 1).

However, as pointed out in [18, 21] the hidden station problem may persist in IEEE 802.11 ad hoc networks even with the use of the RTS/CTS handshake. This is due to the fact that the power needed for interrupting a packet reception is much lower than that of delivering a packet successfully. For more details see the model of the physical layer implemented in NS-2 [22, 23].

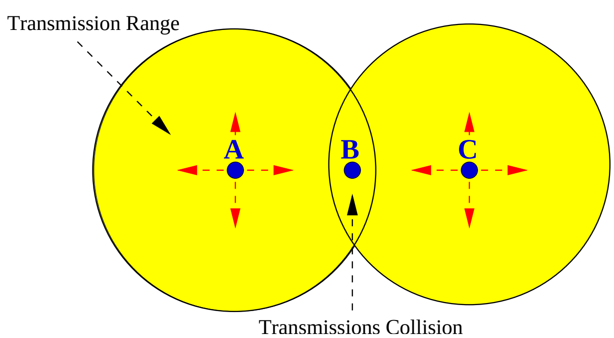


Figure 1: Hidden terminal problem: Packets sent to B by A and C will collide at B.

The exposed station problem results from a situation where a transmission has to be delayed because of the transmission between two other stations within the sender's transmission range. In Figure 2, we show a typical scenario where the exposed terminal problem occurs. Let us assume that A and C are within B's transmission range, and A is outside

C's transmission range. Let us also assume that B is transmitting to A, and C has a frame to be transmitted to D. According to the carrier sense mechanism, C senses a busy channel because of B's transmission. Therefore, station C will refrain from transmitting to D, although this transmission would not cause interference at A. The exposed station problem may thus result in a reduction of channel utilization.

It is worth noting that hidden terminal and exposed terminal problems are correlated with the transmission range. By increasing the transmission range, the hidden terminal problem occurs less frequently. On the other hand, the exposed terminal problem becomes more important as the transmission range identifies the area affected by a single transmission.

Packets transmitted over a fading channel may cause routing protocol to incorrectly conclude that there is a new one-hop neighbor. This one-hop neighbor could provide a shorter route to even more distant nodes. Unfortunately, this new shorter route is usually unreliable. In [24], the authors deploy DSDV [25] and AODV [26] routing protocols in a real network, they find that neither of these protocols can provide a stable route over any multi-hop network connection.

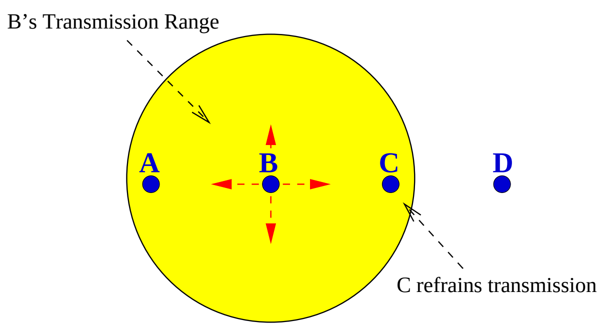


Figure 2: Exposed terminal problem: Because of B's transmission C refrains transmission to D.

## 2.2 Path asymmetry

Path asymmetry in MANETs may appear in several forms like bandwidth asymmetry, loss rate asymmetry, and route asymmetry.

**Bandwidth asymmetry:** Satellite networks suffer from high bandwidth asymmetry, resulting from various engineering tradeoffs (such as power, mass, and volume), as well as the fact that for space scientific missions, most of the data originates at the satellite and flows to the earth. The return link is not used, in general, for data transferring. For example, in broadcast satellite networks a bandwidth ratio of 1000 is not uncommon [10]. On the other hand in MANETs, the degree of bandwidth asymmetry is not very high. For example, the bandwidth ratio lies between 2 and 54 in MANETs that implement the IEEE 802.11 version g protocol [2]. The asymmetry results from the use of different transmission rates. Because of this different transmission rates, even symmetric source destination paths may suffer from bandwidth asymmetry.

**Loss rate asymmetry:** This type of asymmetry takes place when the backward path is significantly more lossy than the forward path. In MANETs, this asymmetry is due to the fact that error-prone depends on local constraints that can vary from place to place. Note that loss rate asymmetry may produce bandwidth asymmetry. For example, in multi-rate IEEE 802.11 protocol versions, senders may use the Auto-Rate-Fallback (ARF) algorithm for transmission rate selection [27]. With ARF, senders attempt to use higher transmission rates after consecutive transmission successes, and revert to lower rates after failures. So, as the loss rate increases the sender will keep using low transmission rates.

**Route asymmetry:** Unlike the previous two forms of asymmetry, where the forward path and the backward path can be the same, route asymmetry implies that distinct paths are used for TCP data and TCP ACKs. This asymmetry may result either from different characteristics of wireless cards or due to the lack of transmission power. If the length of the backward path, expressed in number of hops, is more than the length of the forward path, the backward path will be the bottleneck. Because over a multi-hop wireless path, the throughput decreases rapidly when the number of hops increases [17, 18, 28, 29]. We conclude that route asymmetry may lead to bandwidth asymmetry, and it is up to the routing protocol to select symmetric paths when such routes are available.

In the context of satellite networks, there has been a lot of research on how to improve TCP performance. But since satellite networks are out of the scope of the report, we will limit ourselves to list three techniques introduced by these proposals, which we believe might be useful in MANETs.

The first one is "TCP header compression" that reduces the size of the TCP ACKs on the backward path [30]. The second one is "ACK filtering" that reduces the number of TCP ACKs transmitted, by taking advantage of the fact that TCP ACKs are cumulative [31]. The third one is "ACK congestion control" that let the receiver also control the congestion on the backward path. This is done by dynamically maintaining a delayed-ACK factor $d$ by the receiver, and by sending one ACK for every $d$ data packet received [31]. Unfortunately, these techniques alone cause problems such as increasing sender's burst traffic and also slowing down the sender's congestion window growth. So, it is necessary to adapt the sender congestion control algorithm to avoid these problems. For details about the sender adaptation techniques, we refer to [31];tThe adaptive delayed-ACK proposed in [17] aims to reduce the contention on the channel, by reducing the number of TCP ACKs transmitted. This proposal also alleviates the asymmetry problem in MANETs. We have not found any other proposal dealing with the asymmetry problem in MANETs.

## 2.3 Network partition

An ad hoc network can be represented by a simple graph $G$. Mobile stations are the "vertices". A successful transmission between two mobile stations is an undirected "edge". Network partition happens when $G$ is disconnected. The main reason of this disconnection in MANETs is node mobility. Another factor the can lead to network partition is energy constrained operation of nodes. An example of network partition illustrated in Figure 3. In this figure, when node D moves away from node C this results in a partition of the network into two separate components. Clearly, the TCP agent of node A can not receive the TCP ACK transmitted by node F. If the disconnectivity persists for a duration greater than the retransmission timeout (RTO) of node A, the TCP agent will trigger the *exponential backoff* algorithm [32], which consists of doubling the RTO whenever the timeout expires. Originally, TCP does not have indication about the exact time of network reconnection. This lack of indication may lead to long idle periods during which the network is connected again, but TCP is still in the backoff state.

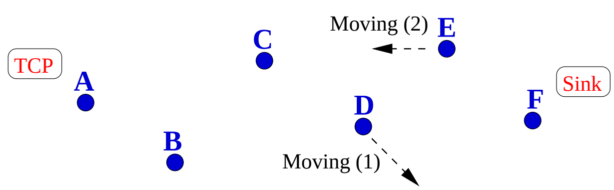


Figure 3: Network partition scenario: When D is moving away from C. The network is reconnected when E is moving toward C.

## 2.4 Routing failures

In wired networks route failures occur very rarely. In MANETs they are frequent events. The main cause of route failures is node mobility. Another factor that can lead to route failures is the repeated transmission failure due to link layer contention. The route re-establishment duration after route failure in MANETs depends on the underlying routing protocol, mobility patterns of nodes, and traffic characteristics. As already discussed in Section 2.3, if TCP sender's does not have indications on the route re-establishment event, the throughput and session delay will degrade because of the large idle time. Also, after the route re-establishment TCP will face a brutal fluctuation in Round Trip Time (RTT) that must be taken into account to compute the new RTO; this is so because during the backoff state TCP does not compute the RTT and increases RTO exponentially [32].

In addition in MANETs, routing protocols that rely on broadcast Hello messages to detect neighbors reachability, may suffer from "communication gray zones" problem. In these zones data messages cannot be exchanged although broadcast HELLO messages and control frames indicate neighbor reachability. In [33] the authors have conducted experiments and have concluded that the origin of this problem was heterogeneous transmission rates, absence of acknowledgment for broadcast packets, small Hello's packet size, and fluctuations of wireless links.

### 2.5 Power constraints

Because batteries carried by each mobile node have limited power supply, processing power is limited. This is a major issue in MANETs, as each node is acting as an end system and as a router at the same time, which the implication that additional energy is required to forward and relay packets. TCP must use this scarse power resource in an "efficient" manner. Here, efficiency means minimizing the number of unnecessary retransmissions at the transport layer as well as at the link layer[3]. In general, in ad hoc networks there are two correlated power problems: the first one is "power saving" that aims at reducing the power consumption; the second one is "power control" that aims at adjusting the transmission power of mobile nodes. Power saving strategies have been investigated at several levels of a mobile device including the physical layer transmissions, the operation systems, and the applications [34]. Power control can be jointly used with routing or transport agents to improve the performance of MANETs [35, 36]; power constraints communications reveal also the problem of cooperation between nodes, as nodes may not participate in routing and forwarding procedures in order to save battery power.

## 3 TCP performance over wireless multi-hop ad hoc networks

In wireless multi-hop ad hoc networks, all TCP performance studies that can be found in the literature are based on simulations and/or experiments. To the best of our knowledge, there does not exist any analytical study aiming at modeling TCP over multi-hop ad hoc networks. In passing, we point out that routing protocols and link layer protocols in MANETs are beyond the scope of this survey. The interested readers are referred to [37, 38] for details.

[3]IEEE 802.11 protocol implements a local retransmission at the link layer upon detecting a transmission error

In [17, 18, 28, 29] the authors report simulation results on TCP throughput in a static linear multi-hop chain, where IEEE 802.11 protocol is used. In Figure 4, we display a multi-hop chain of $N$ nodes. It is expected that, as the number of hops increases, the spatial reuse will also increase. However, simulation results indicate that TCP throughput decreases "rapidly" as the number of hops increases. It is argued that this exponential decrease is due to the hidden terminals problem, which increases packet collisions. After a repeated[4] transmission failure MAC layer will react by first discarding the head-of-line frame destinates to the next hop, and second by notifying the upper layer about a link failure. When the routing protocol of a source node detects a routing failure, it will initiate a route re-establishment process. In general the route re-establishment duration is greater than the retransmission timer of the TCP agent; hence the TCP agent will enter the backoff procedure and will set its congestion window to 1.

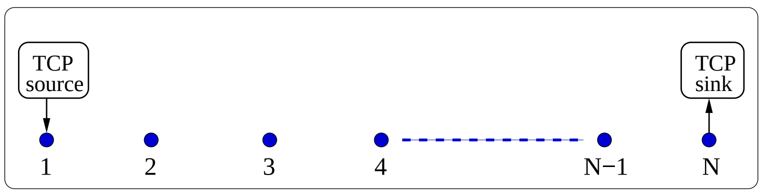


Figure 4: Multi-hop chain topology

In [39], the authors report simulation results on the performance of TCP Reno over three different routing protocols (AODV [26], DSR [40], ADV [41]). It is found that ADV performs well under a variety of mobility patterns and topologies. Further, they propose an heuristic technique called fixed RTO to improve the performance of on-demand routing protocols (AODV and DSR). An overview of this technique is discussed in Section 4.2.1.

In [42], the authors study the performance of TCP Tahoe, Reno, New Reno, Sack and Vegas over the multi-hop chain topology shown in Figure 4, in the case where the IEEE 802.11 protocol is used. It is shown that TCP Vega delivers the better performance and does not suffer from instability. By tuning the sender TCP's maximum window size (advertised window) to approximately four packets, all TCP versions perform similarly. Furthermore, they author investigate the performance of these TCP versions using the delayed-ACK option, as defined in RFC 1122. They report that an improvement of 15% to 32% can be obtained by using this delayed-ACK option. According to the mentioned RFC, the TCP sink will send one TCP ACK packet for every two TCP packet received. In general, the delayed-ACK option has two advantages: first, it reduces the number of TCP ACK packets that contend with the TCP packets to access the channel; second, delayed-ACKs are less aggressive than

[4]On repeated transmission failure, 802.11 MAC is allowed to retransmit short frames seven times and long frames four times.

ACKs sent in the standard TCP [43]. This is so because TCP is "ACK-clocked", namely, the TCP sender increases its congestion window as a function of the "number" of TCP ACK received.

In [44], the authors study the impact of the maximum window size on throughput in MANETs. They prove that regardless of the MAC layer being used, the bandwidth-delay product of multi-hop routes in MANETs cannot exceed the Round-Trip Hop-Count (RTHC) of the path. The bandwidth-delay product is known as the maximum number of packets that the path can support without causing congestion. In case of a multi-hop chain that implements IEEE 802.11 MAC, the bandwidth-delay product cannot exceed one 1/5 of the RTHC. Based on this tighter bound, they propose an algorithm to adjust TCP maximum window size according to the length of the chain. The authors in [18] report that, given a specific network topology and flow patterns, there exists an optimal TCP's window size $W^*$. Using $W^*$ TCP achieves the best throughput via improved spatial reuse. But, unfortunately, TCP operates at an average window size that is much larger than $W^*$; this leads to increased packet loss due to the contention on the wireless channel. To help TCP operating around $W^*$, they propose two techniques: link RED (LRED ) and adaptive pacing at the link layer. By coupling these two techniques they show a 30% improvement in the performance. An overview of these techniques is given in Section 4.2.2.

In [28, 45], the authors study the fairness of TCP in the context of ad hoc network. They report some channel bandwidth unfairness using the actual MAC 802.11 in a multi-hop communication environment. Moreover in [46], the authors show that in scenarios where TCP crosses wireless ad hoc and wired networks, the TCP unfairness problem persists also. The authors in [47] reports that RED [48] did not solve TCP's unfairness in MANETs. The reason is that congestion does not happen in a single node, but in an entire area involving multiple nodes. So, the local packet queue at any single node cannot completely reflect the network congestion state. For this reason, they define a new distributed queue that contains all packets whose transmissions will affect the node transmission in addition to its own packets. An overview of this proposal is given in Section 4.2.2. In [49], the authors investigate the performance of IEEE 802.11 ad hoc network by means of an experimental study. Their findings match those obtained by simulations, namely, TCP connections may experience significant unfairness. They mention several aspects that are usually neglected in simulation studies of IEEE 802.11b protocol. For instance, since the control frames (RTS, CTS, ACK) and data frame may be transmitted at different rates, this produces different transmission ranges and carrier sensing ranges in the network.

# 4 Proposals to improve TCP performance in MANETs

In this section we present the various proposals which have been made in the literature to improve the performance of TCP in MANETs. We classify these proposals in two categories:

cross layer proposals and layered proposals. In layered proposals, the adaptation involves only one OSI layer, whereas in cross layer proposals at least two OSI layers are involved.

## 4.1 Cross layer proposals

Cross layer proposals can be classified in three types: (1) TCP and network cross layer, (2) TCP and physical cross layer, and (3) network and physical cross layer.

### 4.1.1 TCP and network cross layer

**TCP-F:** TCP Feedback [50] is a feedback based approach to handle route failures in MANETs. This approach allows the TCP sender to distinguish between losses due to routes failure and those due to network congestion. This is done as follows. When routing agent of a node detects the disruption of a route, it explicitly sends a Route Failure Notification (RFN) packet to the source. On receiving the RFN, the source goes into a snooze state. TCP sender in snooze state will stop sending packets, and will freeze all its variables, such as timers and congestion window size. The TCP sender remains in this snooze state until it is notified of the restoration of the route through Route Re-establishment Notification (RRN) packet. On receiving the RRN, the TCP sender will leave the snooze state and will resume transmission based on the previous sender window and timeout values. To avoid blocking scenario in the snooze state, the TCP sender, on receiving RFN, triggers a route failure timer. When this timer expires the congestion control algorithm is invoked normally.

The authors report an improvement by using TCP-F over TCP. The simulation scenario is basic and is not based on an ad hoc network. Instead, they emulate the behavior of an ad hoc network from the viewpoint of a transport layer.

**ELFN-based technique:** Explicit Link Failure Notification technique [13] is similar to TCP-F. However in contrast to TCP-F, the evaluation of the proposal is based on a real interaction between TCP and the routing protocol. This interaction aims to inform the TCP agent about route failures when they occur. The authors use an ELFN message, which is piggy-backed on the route failure message sent by the routing protocol to the sender. The ELFN message is like a "host unreachable" Internet Control Message Protocol (ICMP) message, which contains the sender receiver addresses and ports, as well as TCP packet's sequence number. On receiving the ELFN message, the source responds by disabling its retransmission timers and enters a "standby" mode. During the standby period, the TCP sender probes the network to check if the route is restored. If the acknowledgment of the probe packet is received, TCP sender leaves the standby mode, resumes its retransmission timers, and continues the normal operations.

In the mentioned reference, the authors study the effect of varying the time interval between probe packets. Also, they evaluate the impact of the RTO and the Congestion Window (CW) upon restoration of the route. They find that a probe interval of 2 sec.

performs the best, and they suggest to make this interval a function of the RTT instead of giving it a fixed value. For the RTO and CW values upon route restoration, they find that using the prior values before route failure performs better than initializing CW to 1 packet and/or RTO to 6 sec., the latter value being the initial default value of RTO in TCP Reno and New Reno versions.

This technique provides significative enhancements over standard TCP, but further evaluations are still needed. For instance, different routing protocols should be considered other than the reactive protocol DSR considered in [13], especially proactive protocols such as OLSR [51]. Also, values other than 2 sec. for the probe interval should be checked as well.

**ATCP:** Ad hoc TCP [14] utilizes network layer feedback too. In addition to the route failures, ATCP tries to deal with the problem of high Bit Error Rate (BER). The TCP sender can be put into persist state, congestion control state or retransmit state. A layer called ATCP is inserted between TCP and IP layers of the TCP source node's. ATCP listens to the network state information provided by ECN (Explicit Congestion Notification) messages [52] and by ICMP "Destination Unreachable" message; then ATCP puts TCP agent into the appropriate state. On receiving a "Destination Unreachable" message, TCP agent enters a persist state. The TCP agent during this state is frozen and no packets are sent until a new route is found by probing the network. The ECN is used as a mechanism to explicitly notify the sender about network congestion along the route being used. Upon reception of ECN, TCP congestion control is invoked normally without waiting for a timeout event. To detect packet losses due channel errors, ATCP monitors the received ACKs. When ATCP sees that three duplicate ACKs have been received, it does not forward the third duplicate ACK but puts TCP in the persist state and quickly retransmits the lost packet from TCP's buffer. After receiving the next ACK, ATCP will resume TCP to the normal state. Note that ATCP allows interoperability with TCP sources or destinations that do not implement ATCP.

ATCP was implemented in a testbed and evaluated under different scenarios, such as congestion, lossy links, partition, and packet reordering. In all cases the transfer time of a given file using ATCP yielded better performance comparatively to TCP. However, the used scenario was somewhat special, since neither wireless links nor ad hoc routing protocols were considered. In fact, the authors used an experimental testbed consisting of five PCs equipped with Ethernet cards. With these PCs, the authors formed a four hop network. Some assumptions such as ECN-capable node as well as sender node being always reachable might be somehow hard to meet in mobile ad hoc context.

**TCP-BuS:** TCP Buffering capability and Sequence information [16], like previous proposals, uses the network feedback in order to detect route failure events and to take convenient reaction to this event. The novel scheme in this proposal is the introduction of *buffering* *capability* in mobile nodes. The authors select the source-initiated on-demand ABR [53] (Associativity-Based Routing) routing protocol. The following enhancements are proposed:

- Explicit notification: two control messages are used to notify the source about the route failure and the route re-establishment. These messages are called Explicit Route Disconnection Notification (ERDN) and Explicit Route Successful Notification (ERSN). On receiving the ERDN from the node that detected the route failure, called the Pivoting Node (PN) , the source stops sending. And similarly after route re-establishment by the PN using a Localized Query (LQ), the PN will transmit the ERSN to the source. On receiving the ERSN, the source resumes data transmission.

- Extending timeouts values: during the Route ReConstruction (RRC) phase, packets along the path from the source to the PN are buffered. To avoid timeout events during the RRC phase, the retransmission timer value for buffered packets is doubled.

- Selective retransmission request: as the retransmission timer value is doubled, the lost packet along the path from the source to the PN are not retransmitted until the adjusted retransmission timer expires. To overcome this, an indication is made to the source so that it can retransmit these lost packet selectively.

- Avoiding unnecessary requests for fast retransmission: when the route is restored, the destination notifies the source about the lost packets along the path from the PN to the destination. On receiving this notification, the source simply retransmits these lost packets. But the packets buffered along the path from the source to the PN may arrive at the destination earlier than the retransmitted packets. So. the destination will reply by duplicate ACK. These unnecessary request packets for fast retransmission are avoided.

- Reliable retransmission of control message: in order to guarantee the correctness of TCP-BuS operation, they propose to transmit reliably the routing control messages ERDN and ERSN. The reliable transmission is done by overhearing the channel after transmitting the control messages. If a node has sent a control message but did not overhear this message relayed during a timeout, it will conclude that the control message is lost and it will retransmit this message.

This proposal introduces many new techniques for TCP's improvement. The novel contributions of this paper are the buffering techniques and the reliable transmission of control messages. In their evaluation, the authors found that TCP-BuS outperforms the standard TCP and the TCP-F under different conditions. The evaluation is based only on the ABR routing protocol and different routing protocol should be taken into account.

**Split TCP:** TCP connections that have large number of hops suffer from frequent route failures due to mobility. To improve the throughput of these connections and to resolve the unfairness problem, the Split TCP scheme was introduced to split long TCP connections into shorter localized segments [54] – see Figure 5. The interfacing node between two localized segments is called proxy. The routing agent decides if its node has the role of proxy according to the inter-proxy distance parameter. The proxy intercepts TCP packets, buffers them, acknowledges their receipt to the source (or previous proxy) by sending a local acknowledgment (LACK). Also, a proxy is responsible for delivering the packets, at an appropriate rate, to the next local segment. Upon the receipt of a LACK (from the next proxy or from the final destination), a proxy will purge the packet from its buffer. To ensure the source to destination reliability, an ACK is sent by the destination to the source similarly to the standard TCP. In fact, this scheme splits also the transport layer functionalities into those end-to-end reliability and congestion control. This is done by using two transmission windows at the source which are the congestion window and the end-to-end window. The congestion window is a sub-window of the end-to-end window. While the congestion window changes in accordance with the rate of arrival of LACKs from the next proxy, the end-to-end window will change in accordance with the rate of arrival of the end-to-end ACKs from the destination. At each proxy, there would be a congestion window that would govern the rate of sending between proxies.

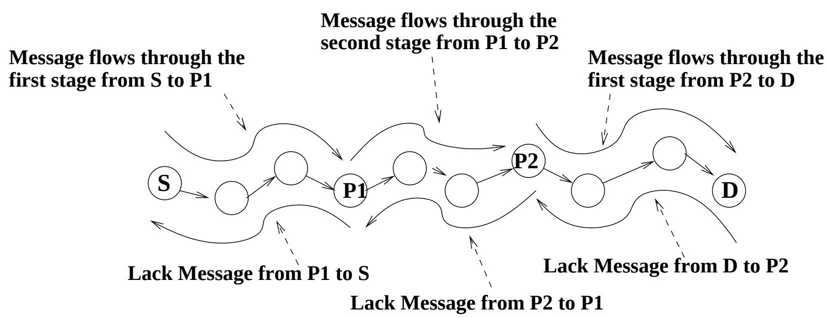


Figure 5: TCP Split

Simulation results indicate that an inter-proxy distance between 3 and 5 has a good impact on both throughput and fairness. The authors report that an improvement up to 30% can be achieved in the total throughput by using Split TCP. The drawbacks are large buffers and network overhead.

### 4.1.2 TCP and physical cross layer

**Jointly Optimal Congestion-control and Power-control [36]:** Power control in the physical layer can often influence the transmission rate of mobile nodes. For example, mobile nodes, that implement the IEEE 802.11 protocol and the ARF rate selection algorithm, reduce their transmission rate in case of packets loss. These losses may result from low signal to interference power ratio.

In MANETs, the joint congestion and power control problem is to maximize the sources utility function $U_s$ subject to link capacity constraints $c_l$. The sources utility function is represented through continuously differentiable, increasing, and strictly concave function of the source's transmission rate $x_r$. The author considers Code Division Multiple Access (CDMA) ad hoc networks. In this type of networks, the capacity of a link $l$ is not fixed, as it appears as a logarithmic function of signal to interference ratio ($SIR$). To solve this optimization problem, the author proposes a distributed algorithm called Jointly Optimal Congestion-control and Power-control (JOCP). The key idea of JOCP is that, during congestion periods, nodes will try to transmit packets faster at the bottleneck links by updating their transmission power. More specifically, at each time slot the transmission power at a transmitter $i$ will increase proportionally to its packet queuing delay $\lambda_i$ and will decrease proportionally to its current power level $P_i$. Also, its transmission power should depend of other nodes' queuing delay and $SIR$. For this reason, The authors makes the transmission power decreases by a weighted sum of the variables $m_j$ received from all other node $j$ through a flooding protocol, where

$$m_j = \frac{\lambda_j(t)SIR_j(t)}{P_j(t)G} \tag{1}$$

with $\lambda_j$ is the packet queuing delay at node $j$, $SIR_j$ the signal to interference ratio at node $j$, $P_j$ is the j's transmission power, and $G$ is the path loss. Moreover, each time slot $t$, the TCP window size is updated using the total propagation delay between the source and destination $D_s$, like in TCP Vegas version. JOCP converges geometrically fast and its convergence can be maintained under any finite clock asynchronism. The author reports that JOCP is robust to wireless channel variations and path loss estimation errors. By expanding the scope of network utility function to handle elastic link capacities, the author proves that JOCP achieves optimal balance between transport and physical layers in wireless CDMA ad hoc networks.

### 4.1.3 Network and physical cross layer

**Preemptive routing in ad hoc networks:** This objective of this proposal is to reduce the number of routing failures [55]. This is achieved by switching to a new route when a link of the current route is expected to fail in the future (see below). This technique is coupled with the on-demand routing protocol AODV and DSR. The detection mechanism of failure is power based. More specifically, when the signal power drops below a given, called preemptive threshold, the routing agent of the source is notified (see Figure 6). On receiving this notification, the source's routing agent proactively looks up for a new route. When the new route is available, the routing agent switches to this new route. The value of the preemptive threshold appears to be critical. Indeed, in case of low threshold value, there will not be sufficient time to discover an alternative path before the route fails. Also, in the case of high value, the warning message will be generated too early. To overcome the fluctuations of the received signal power due to channel fading and multipath effects, that may trigger a preemptive route warning and cause unnecessary route request floods, the authors use a repeated short message probing to verify the correctness of the warning message.

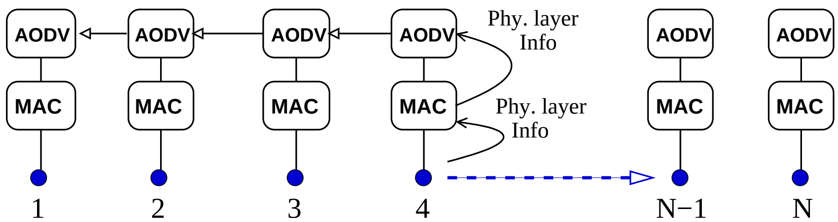


Figure 6: Network and Physical cross layer

Using simulations the authors show that their scheme yields a reduction of the number of route failures and decreases the latency by 30%. It should be noted that this scheme is "packet receipt event-driven" and that failures cannot be detected if no packets are transmitted.

**Signal strength based link management in ad hoc network:** This algorithm [35] is similar to the previous one. However, in this algorithm each node keeps a record of the received signal strengths of 1-hop neighboring nodes. Using these records, the routing protocol predicts link break event in the immediate future[5]; this prediction is called Proactive Link Management. On detecting this event, the source's routing agent is notified by a Going Down message – see Figure 6. On receiving this message the source's routing agent stops sending packets, and initiates a route discovery procedure. The novelty of this proposal is the Reactive Link Management mechanism. This mechanism increases the transmission power to re-establish a broken link. Reactive and Proactive Link Management mechanisms can be coupled in the following way: on predicting that a link is going to be down, the node's routing agent notifies the source to stop sending, and this node increases its transmitting power to handle packets in transit that use this link.

The authors use simulations to show that their scheme yields 45% improvement in TCP performance. Note that only light load scenarios are considered.

[5] Immediate future means after 0.1 sec..

## 4.2 Layered proposals

We classify layered proposals according to which layer the adaptation is done: at the TCP layer or at the link layer.

### 4.2.1 TCP layer proposals

**Fixed RTO:** This technique [39] is a sender-based technique that does not rely on feedback from the network. In fact, the authors employ a heuristic to distinguish between route failures and congestion. When two timeout expire in sequence, which corresponds to the situation where the missing ACK is not received before the second RTO expires, the sender concludes that a route failure event has occured. The unacknowledged packet is retransmitted but the RTO is not doubled a second time. This is in contrast with the standard TCP, where an "exponential" backoff algorithm is used. The RTO remains fixed until the route is re-established and the retransmitted packet is acknowledged.

In [39] the authors evaluate this proposal by considering different routing protocols as well as the TCP selective and the delayed acknowledgment options. They report that significant enhancements are achieved when using fixed-RTO with on-demand routing protocols. Nevertheless, as stated by the authors themselves, this proposal is restricted to wireless networks only, a serious limitation since interoperation with wired networks is clearly necessary. Also, the supposition that two consecutive timeouts are the exclusive results of route failures need more analysis, especially in cases of congestion.

**TCP DOOR:** TCP Detection of Out-Of-Order and Response (DOOR) is an end-to-end approach [15]. This approach, which does not require the cooperation of intermediate nodes, is based on out-of-order (OOO) delivery events. OOO events are interpreted as an indication of route failure. The detection of OOO events is accomplished either by the means of a sender-based or a receiver-based mechanism. The sender-based technique uses the non-decreasing property of the ACKs sequence number to detect the OOO events. In case of duplicate ACK packets, these ACKs will have the same sequence number, so that the sender needs additional information to detect OOO event. This information is a one byte option added to ACKs called ACK Duplication Sequence Number (ADSN). The ADSN is incremented and transmitted with each duplicate ACK. However, the receiver needs an additional two bytes TCP option to detect OOO events, called TCP Packet Sequence Number (TPSN). The TPSN is incremented and transmitted with each TCP packets including the retransmitted packets. If the receiver detects an OOO event, it should notify the sender by setting a specific option bit, called OOO bit, in the ACK packet header.

Once the TCP sender knows about an OOO event, it takes the following two response actions: temporarily disabling congestion control, and instant recovery during congestion avoidance. In the former action, the TCP sender disables the congestion algorithm for a specific time period ($T_1$). In the latter action, if the congestion control algorithm was invoked during the past time period ($T_2$), the TCP sender should recover immediately to the state before the invocation of the congestion control. In fact, the authors make the time periods $T_1$ and $T_2$ function of the RTT.

In the simulation study presented in [15], different scenarios are considered by combining all mechanisms and actions mentioned above. Their results show that the sender and receiver-based mechanisms behave similarly. So, they recommend the use of the sender detection mechanism as it does not require notifications from the sender to the receiver. Regarding the actions, temporarily disabling congestion control and instant recovery during congestion avoidance, to be taken upon an OOO event detection, they have found that both of them lead to significant improvement. In general, TCP DOOR improves TCP performance up to 50%. Nevertheless, the supposition that OOO events are the exclusive results of route failure deserves much more analysis. Actually, multipath routing protocols such as TORA [56] may produce OOO events that are not related to route failures.

**Dynamic delayed Ack:** This approach [17] aims to reduce the contention on wireless channel, by decreasing the number of TCP ACKs transmitted by the sink. It is a modification of the delayed ACK option (RFC 1122) that has a fixed coefficient $d = 2$. In this approach, the value of $d$ varies with the sequence number of the TCP packet. The authors define three thresholds $l1$, $l2$, and $l3$ such that $d = 1$ for packets with sequence number $N$ smaller than $l1$, $d = 2$ for packets with $l1 \leq N \leq l2$, $d = 3$ for $l2 \leq N \leq l3$ and $d = 4$ for $l3 \leq N$. They report simulation results for a chain ad hoc topology. They show that their proposal outperforms the standard TCP as well as the delayed ACK option for a fixed coefficient $d = 2, 3, 4$. In their analysis, they study the packet loss rate, throughput and session delay of TCP New Reno, in the case of short and persistant TCP sessions. They suggest that better performance could be obtained by making $d$ a function of the sender's congestion window instead of a function of the sequence number.

**Adaptive CWL setting:** In [44], the authors study the bandwidth delay product (BWD) of multi-hop routes in MANETs. They show that BWD cannot exceed the Round-Trip Hop-Count (RTHC) of the path. In the case of a multi-hop chain that implements IEEE 802.11 MAC, they find that the BWD is tighter and is equal to 1/5 the RTHC.Based on this tighter bound, they propose to use the adaptive CWL setting algorithm to adjust TCP's maximum window size according to the length of the chain. They report that using this algorithm an improvement up to 16% can be achieved.

### 4.2.2 Link layer proposals

**Link RED** :Link RED (LRED) [18] aims to reduce the contention on the wireless channel. This is done by monitoring the average number of retries (avg) in the packet transmission at the link layer. When *avg* number becomes greater than a given threshold, the probability of dropping/marking is computed according to the RED algorithm [48]. Since it marks packets, LRED can be coupled with Explicit Congestion Notification (ECN) in order to notify the TCP sender about the congestion level [52]. However, instead of notifying the TCP sender about congestion level, the authors increase the backoff time at the MAC layer. The fairness between TCP connections using this technique should be analyzed, as in this study only the impact of this technique on TCP throughput is discussed.

**Adaptive pacing** :The goal of this proposal [18] is to improve spatial channel reuse. In the current IEEE 802.11 protocol, a node is constrained from contending for the channel by a random backoff period, plus a single packet transmission time that is announced by RTS or CTS frame. However, the exposed receiver problem persists due to the lack of coordination between nodes that are two hops away from each other. Adaptive pacing solves this problem by increasing the backoff period by an additional packet transmission time. This proposal works together with LRED as follows. Adaptive pacing is enabled by LRED. When a node finds its average number of transmission retries to be less than a threshold, it calculates its backoff time as usual. When the average number of retries goes beyond this threshold, adaptive pacing is enabled and the backoff period is increased by a duration equal to the transmission time of the previous packet. Again, the impact of these techniques on TCP fairness should be analyzed, as well as the delay of short TCP sessions. Also in Adaptive pacing, as we mentioned, the additional backoff time is based on the size a packet, so the existence of different data packets size in the network should be inspected.

**Non work-conserving scheduling:** The goal of this proposal [57] is to improve fairness among TCP flows crossing wireless ad hoc and wired networks. The authors adopt the "non work-conserving scheduling" policy for ad hoc networks instead of the "work-conserving scheduling". This done as follows. The link layer queue[6] sets a timer each time it sends a data packet to the MAC. Only after timer expiration, the queue outputs another packet to the MAC. The duration of timer is updated according to the queue output rate value. Specifically, the duration of the timer is a sum of three parts $D_1$, $D_2$, and $D_3$. $D_1$ is equal to the data packet length divided by the bandwidth of the channel. $D_2$ is a delay, the value of which is decided by the recent queue output rate. The queue calculates the output rate by counting the number of bytes, $C$, it outputs in every fixed interval $T$. To decide on the value of $D_2$, the authors set three thresholds $X$, $Y$, and $Z$ ($X < Y < Z$) for C, and four delay values $D_{21}$, $D_{22}$, $D_{23}$, and $D_{24}$ ($D_{21} < D_{22} < D_{23} < D_{24}$) for $D_2$, as shown in (2). The heuristic behind (2) is to penalize greedy nodes with high output rate by increasing their queueing delay $D_2$ and to favor nodes t with small output rates. $D_3$ is a random value uniformly distributed between 0 and $D_2$. $D_3$ is used to avoid synchronization problem and to reduce collisions. By means of simulations, the authors report that their scheme greatly improves the fairness among TCP connections at the cost of moderate total throughput degradation.

[6] In NS-2, Link layer queue is called InterFace queue (IFq).

$$\begin{cases} D_2 = D_{21} & \text{for } C \leq X \\ D_2 = D_{22} & \text{for } X < C \leq Y \\ D_2 = D_{23} & \text{for } Y < C \leq Z \\ D_2 = D_{24} & \text{for } C > Z \\ 0 \leq D_{21} < D_{22} < D_{23} < D_{24} & \end{cases} \tag{2}$$

**Neighborhood RED:** This proposal [47] aims to enhance TCP fairness in MANETs. Unlike in wired networks, the authors show that RED does not solve TCP's unfairness in MANETs, because the congestion does not happen in a single node, but in an entire area involving multiple nodes. The local packets queue at any single node cannot completely reflect the network congestion state. For this reason, the authors define a new distributed queue, called neighborhood queue. At a node, the neighborhood queue should contain all packets whose transmissions will affect its own transmission in addition to its packets[7]. Since it is difficult to get information about all these packets without introducing significant communication overhead, which may need 2-hop information exchange, a simplified node neighborhood queue is introduced. It aggregates the node's local queue, and the upstream and downstream queues of its 1-hop neighbors. Now, the RED algorithm is based on the average queue size of the neighborhood queue. The authors use a distributed algorithm to compute the average queue size. In this algorithm, the time is slotted. During each time slot, the idle period of the channel is measured. Using this measurement, a node estimates the channel utilization and the average neighborhood queue size. The accuracy of the estimation is controlled by the duration of the slots. Using simulations, they verify the effectiveness of their proposal and the fairness improvement of TCP.

# 5 Conclusion

We have presented a state-of-the-art of TCP over Mobile Ad hoc Networks (MANETs). The principal problem of TCP in this environment is clearly its inability to distinguish between losses induced by network congestion and others types of losses. TCP assumes that losses are always due to network congestion. But while this assumption is valid, in most cases, in wired networks, it is not true in MANETs. In MANETs, there are indeed several types of losses, including losses caused by routing failures, by network partitions and by high bit error rates. Performing congestion control in these cases, like TCP does, yields poor performance. In order to solve this problem, several proposals have been made to notify TCP about the cause of a loss. The difference between these proposals lies in how notifications are done and how to react. We have classified these proposals as layered and cross layer. In cross layer proposals, TCP and the underlying protocols cooperate to improve MANET performance. For example, TCP and Network cross layer proposals use explicit notification, which requires

[7] Affect means : prevent a node from transmitting because of channel capturing, or to induce transmission failure to carrier sensing MAC protocols.

cooperation from intermediate nodes like in TCP-F, TCP-ELFN, ATCP, TCP-BuS. In layered proposals, one OSI layer is adapted. For example, TCP layer proposals require only the cooperation of the sender and receiver, like in TCP-DOOR and Fixed-RTO. However, cross layer proposals report higher improvement than layered ones.

The frequent route failures and route re-establishments in MANET environment introduce a new challenge to TCP congestion control algorithm, and leads us to pose the following questions: does the actual congestion control algorithm behave efficiently in dynamic environments such as MANETs? Are the parameters values taken in TCP to compute the retransmission timeout after a route re-establishment valid in MANETs? So, as an extension of this work we intend to answer these questions.

## Aknowledegments

We would like to thank Daniele Miorandi and Urtzi Ayesta for their help. This work was supported in part by the Euro NGI network of excellence.

## References

[1] "Bluetooth Special Interest Group," Web site: http://www.bluetooth.com.

[2] "IEEE 802.11 WLAN standard," Web site: http://standards.ieee.org/getieee802.

[3] "Broadband radio access networks (BRAN): High performance Local Area Network (HiperLAN) type 2," Tech. Rep. 101 683 V1.1.1, ETSI.

[4] L. Andrew, S. Hanly, and R. Mukhtar, "CLAMP: Differentiated capacity allocation in access networks," in *Proc. of IEEE Int. Performance Computing and Communications Conf.*, Phoenix, AZ, USA, Apr. 2003, pp. 451–458.

[5] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 756–769, Dec. 1997.

[6] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz, "Improving TCP/IP performance over wireless networks," in *Proc. of ACM MOBIHOC*, Berkeley, CA, USA, 1995, pp. 2–11.

[7] A. V. Bakre and B.R. Badrinath, "Implementation and performance evaluation of indirect TCP," *IEEE/ACM Transactions on Networking*, vol. 46, no. 3, pp. 260–278, Mar. 1997.

[8] K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 5, pp. 19–43, Oct. 1997.

[9] H. Balakrishnan, S. Seshan, and R. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," *ACM Wireless Networks*, vol. 1, no. 4, pp. 469–481, Dec. 1995.

[10] R. Durst, G. Miller, and E. Travis, "TCP extensions for space communications," in *Proc. of ACM MOBICOM*, Rye, NY, USA, 1996, pp. 15–26.

[11] T. Henderson and R. Katz, "Transport protocols for Internet-compatible satellite networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 2, pp. 345–359, Feb. 1999.

[12] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A Feedback based scheme for improving TCP performance in ad hoc wireless networks," in *Conference on Distributed Computing Systems*, Amsterdam, Netherlands, May 1998, pp. 472–479.

[13] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," *ACM Wireless Networks*, vol. 8, no. 2, pp. 275–288, Mar. 2002.

[14] J. Liu and S. Singh, "ATCP: TCP for mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 7, pp. 1300–1315, Jul. 2001.

[15] F. Wang and Y. Zhang, "Improving TCP performance over mobile ad hoc networks with out-of-order detection and response," in *Proc. of ACM MOBIHOC*, Lausanne, Switzerland, Jun. 2002, pp. 217–225.

[16] D. Kim, C. Toh, and Y. Choi, "TCP-BuS: Improving TCP performance in wireless ad hoc networks," *Journal of Communications and Networks*, vol. 3, no. 2, pp. 175–186, Jun. 2001.

[17] E. Altman and T. Jiménez, "Novel delayed ACK techniques for improving TCP performance in multihop wireless networks," in *Proc. of the Personal Wireless Communications*, Venice, Italy, Sep. 2003, pp. 237–253.

[18] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP throughput and loss'," in *Proc. of IEEE INFOCOM*, San Francisco, USA, Apr. 2003.

[19] F. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part ii - the hidden terminal problem in Carrier Sense Multiple-Access modes and the busy-tone solution," *IEEE Transactions on Communications*, vol. 23, no. 12, pp. 1417–1433, 1975.

[20] V. Bharghavan, A. Demers, S. Shneker, and L. Zhang, "MACAW: a media access protocol for wireless LAN's," in *Proc. of ACM SIGCOMM*, London, UK, 1994, pp. 212–225.

[21] K. Xu, M.Gerla, and S. Bae, "Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks," *Ad Hoc Networks Journal, Elsevier*, vol. 1, no. 1, pp. 107–123, Jul. 2003.

[22] "The Network Simulator NS-2," Web site: http://www.isi.edu/nsnam/ns/index.html.

[23] "Global Mobile Information Systems Simulation Library GloMoSim," Web site: http://pcl.cs.ucla.edu/projects/glomosim/.

[24] K. Chin, J. Judge, A. Williams, and R. Kermode, "Implementation experience with MANET routing protocols," *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 5, pp. 49–59, Nov. 2002.

[25] C. Perkins and T. Watson, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," in *Proc. of ACM SIGCOMM*, London, UK, 1994.

[26] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, Category: Experimental, Jul. 2003.

[27] A. Kamerman and L. Monteban., "Wavelan ii: A high-performance wireless lan for the unlicensed band," *Bell Labs Technical Journal*, pp. 118–133, Summer 1997.

[28] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?," *IEEE Communications Magazine*, vol. 39, no. 6, pp. 130–137, Jun. 2001.

[29] M. Gerla, K. Tang, and R. Bagrodia, "TCP performance in wireless multi-hop networks," in *Proc. of the IEEE WMCSA*, New Orleans, LA, USA, 1999.

[30] V. Jacobson, "Compression TCP/IP headers for low speed serial links," RFC 1144, Category: Proposed Standard, Feb. 1990.

[31] H. Balakrishnan and V. Padmanabhan, "How network asymmetry affects TCP," *IEEE Communications Magazine*, pp. 2–9, Apr. 2001.

[32] V. Paxson and M. Allman, "Computing TCP's retransmission timer," RFC 2988, Category: Standard Track, Nov. 2000.

[33] H. Lundgren, E. Nordstro, and C. Tschudin, "Coping with communication gray zones in IEEE 802.11b based ad hoc networks," in *Proc. of the ACM Workshop on Wireless Mobile Multimedia*, Atlanta, GA, USA, Sep. 2002, pp. 49–55.

[34] C. Jones, K. Sivalingam, P. Agarwal, and J. Chen, "A survey of energy efficient network protocols for wireless and mobile networks," *ACM Wireless Networks*, vol. 7, no. 4, pp. 343–358, 2001.

[35] F. Klemm, S. Krishnamurthy, and S. Tripathi, "Alleviating effects of mobility on tcp performance in ad hoc networks using signal strength based link management," in *Proc. of the Personal Wireless Communications*, Venice, Italy, Sep. 2003, pp. 611–624.

[36] M. Chiang, "Balancing transport and physical layers in wireless ad hoc networks: jointly optimal TCP congestion coontrol and power control," *Submitted to IEEE Journal on Selected Areas in Communications.*

[37] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Journal of Ad Hoc Networks, Elsivier*, vol. 2, no. 1, pp. 1–22, Jan. 2004.

[38] I. Chlamtac, M. Conti, and J. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad Hoc Networks Journal, Elsevier*, vol. 1, no. 1, pp. 13–64, Jul. 2003.

[39] T. Dyer and R. Boppana, "A comparison of TCP performance over three routing protocols for mobile ad hoc networks," in *Proc. of ACM MOBIHOC*, Long Beach, CA, USA, 2001, pp. 56–66.

[40] D. Johnson, D. Maltz, and Y. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," Internet draft, Apr. 2003.

[41] R. Boppana and S. Konduru, "An adaptive distance vector routing algorithm for mobile, ad hoc networks," in *Proc. of IEEE INFOCOM*, Anchorage, Alaska, USA, Apr. 2001.

[42] S. Xu and T. Saadawi, "Performance evaluation of TCP algorithms in multi-hop wireless packet networks," *Journal of Wireless Communications and Mobile Computing*, vol. 2, no. 1, pp. 85–100, 2002.

[43] M. Allman, "On the generation and use of TCP acknowledgments," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 5, pp. 4–21, Oct. 1998.

[44] K. Chen, Y. Xue, and K. Nahrstedt, "On setting TCP's congestion window limit in mobile Ad Hoc networks," in *Proc. of IEEE ICC*, Anchorage, Alaska, USA, May 2003.

[45] K. Tang and M. Gerla, "Fair sharing of MAC under TCP in wireless Ad Hoc networks," in *Proc. of IEEE Multiclass Mobility and Teletraffic for Wireless Communications Workshop*, Venice, Italy, Oct. 1999.

[46] K. Xu, S. Bae, S. Lee, and M. Gerla, "TCP behavior across multihop wireless networks and the wired internet," in *Proc. of the ACM Workshop on Wireless Mobile Multimedia*, Atlanta, GA, USA, Sep. 2002, pp. 41–48.

[47] K. Xu, M. Gerla, L. Qi, and Y. Shu, "Enhancing TCP fairness in ad hoc wireless networks using neighborhood red," in *Proc. of ACM MOBICOM*, San Diego, CA, USA, Sep. 2003, pp. 16–28.

[48] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[49] G. Anastasi, M. Conti, and E. Gregori, "IEEE 802.11 ad hoc networks: performance measurements," in *Proc. of the Workshop on Mobile and Wireless Network (MWM 2003) in conjuction with ICDCS 2003*, May 2003.

[50] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A feedback based scheme for improving TCP performance in Ad-Hoc wireless networks," in *Proc. of the International Conference on Distributed Computing Systems (ICDCS'98)*, Amsterdam, Netherlands, May 1998.

[51] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626, Category: Experimental, Oct. 2003.

[52] K. Ramakrishnan, S. Floyd, and D. Black, "The addition of explicit congestion notification (ECN) to IP," RFC 3168, Category: Standards Track, Sep. 2001.

[53] C. Toh, "Associativity-based routing for ad hoc mobile networks," *Journal of Wireless Personal Communications*, vol. 4, no. 2, pp. 103–139, Mar. 1997.

[54] S. Kopparty, S. Krishnamurthy, M. Faloutous, and S. Tripathi, "Split TCP for mobile ad hoc networks," in *Proc. of IEEE GLOBECOM*, Taipei, Taiwan, Nov. 2002.

[55] T. Goff, N. Abu-Ghazaleh, D. Phatak, and R. Kahvecioglu, "Preemptive routing in ad hoc networks," in *Proc. of ACM MOBICOM*, Rome, Italy, 2001, pp. 43–52.

[56] C. Perkins, *AD HOC Networking*, Adison-Wesly, Upper Saddle River, NJ, USA, 2001.

[57] L. Yang, W. Seah, and Q. Yin, "Improving fairness among TCP flows crossing wireless ad hoc and wired networks," in *Proc. of ACM MOBIHOC*, Annapolis, Maryland, USA, Jun. 2003, pp. 57–63.

# Abbreviations and acronyms

| | |
|---|---|
| ABR | Associativity Based Routing protocol |
| ACK | Acknowledgment |
| ADSN | Acknowledgment Duplication Sequence Number |
| ADV | Adaptive Distance Vector |
| AODV | Ad hoc On-demand Distance Vector |
| ARF | Automatic Rate Fallback |
| ARQ | Automatic Repeat Request |
| ATCP | Ad hoc TCP Proposal |
| BWD | Bandwidth Delay Product |
| CDMA | Code Division Multiple Access |
| CTS | Clear To Send |
| CW | TCP Congestion Window |
| CWL | Maximum TCP Transmission Window |
| DCF | Distributed Coordination Function |
| DSDV | Dynamic destination Sequenced Distance Vector |
| DSR | Dynamic Source Routing |
| ECN | Explicit Congestion Notification |
| ELFN | Explicit Link Failure Notification |
| ERDN | Explicit Routing Disconnection Notification |
| ERSN | Explicit Routing Successful Notification |
| FEC | Forward Error Correction |
| ICMP | Internet Control Message Protocol |
| IFq | Link Layer Interface Queue |
| IP | Internet Protocol |
| JOCP | Jointly Optimal Congestion Control and Power Control |
| LACK | Local Acknowledgment |
| LAN | Local Area Network |
| LQ | Localized Query |
| LRED | Link Layer Random Early Detection |
| MAC | Medium Access Control |
| MANETs | Mobile Ad Hoc Networks |
| OLSR | Optimized Link State Routing |
| OOO | Out-Of-Order Event |
| PN | Pivoting Node |
| RED | Random Early Detection |
| RRC | Route Reconstruction |
| RTHC | Round Trip Hop Count |

| | |
|---|---|
| RTO | Retransmission TimeOut |
| RTS | Ready To Send |
| RTT | Round Trip Time |
| SNR | Signal to Noise Ratio |
| TCP | Transmission Control Protocol |
| TCP-BuS | TCP Buffering Capability and Sequence Information |
| TCP-DOOR | TCP Detection Of Out-Of-Order and Response |
| TCP-F | TCP Feedback Proposal |
| TORA | Temporally Ordered Routing Algorithm |
| TPSN | TCP Packet Sequence Number |
| WLAN | Wireless Local Area Network |

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399